ATTY. DOCKET 42390P10363 PATENT

APPLICATION FOR UNITED STATES LETTERS PATENT

FOR

METHOD AND SYSTEM FOR PARAMETERIZED NORMAL PREDICTIVE ENCODING

Inventor: Francis M.L. Ng

Prepared by:
Blakely, Sokoloff, Taylor & Zafman
1279 Oakmead Parkway
Sunnyvale, California 94086
(408) 720-8300

"Express Mail" mailing label number EL627469519US
Date of Deposit December 29, 20000
I hereby certify that this paper or fee is being deposited with the United States Postal Service "Express Mail Post Office to Addressee" service under 37 CFR 1.10 on the date indicated above and is addressed to the Assistant Commissioner of Patents and Trademarks, Washington, D.C. 20231.

Lucia Sanchez
(Typed or printed name of person mailing paper or fee)

(Signature of person mailing paper or fee)

METHOD AND SYSTEM FOR PARAMETERIZED NORMAL PREDICTIVE ENCODING

FIELD OF THE INVENTION

[001] This invention relates to computer graphics, and more specifically to a method and system that efficiently encode the normals of three-dimensional graphics models.

BACKGROUND OF THE INVENTION

[002] Surface normal is an indispensable component of geometric data in the rendering of 3D graphics models. Accurately calculated normal can greatly enhance the perceived quality of 3D graphics models. Normals are customarily represented in three 32-bit IEEE floating-point numbers. In state-of-the-art high-end graphics rendering, each frame can easily contain 1 million triangles with frame rate about 60 frames/second. Assuming that a normal is associated with each vertex of a triangle (4 bytes/coordinate x 3 coordinate/vertex x 3 vertices/triangle=36 bytes/triangle), the data rate for uncompressed normal alone can be as much as 2.18 GB/s (ignoring vertex coordinates, colors and texture coordinates). The typical memory bandwidth between main memory and CPU as of today ranges from 800 MB/s to 1 GB/s (AGP 4x). The memory bus bandwidth is a major bottleneck in three-dimensional graphics rendering. Compressing normal is just as important as compressing other geometric data such as vertex coordinates, colors and texture coordinates in cutting down the high data rate of 3D graphics models.

[003] A group known as the Motion Picture Expert Group has been defining an image compression system known as "MPEG4" that compresses images by defining objects in the image with three-dimensional object models. The compressed images can be reconstructed by rendering the scenes using the three-dimensional object models. In MPEG4 standard, a 3D model coding (3DMC) technique is proposed to reduce the data size of the three-dimensional object models. The technique consists of two main components: connectivity encoding and geometry encoding. Connectivity encoding reduces the redundancy inherent to the representations of polyhedral or triangular 3D meshes. For example, one straightforward but inefficient technique to represent a mesh is to represent each triangle in a mesh by direct reference to its 3 vertices. 3DMC first decomposes a mesh into spanning trees of triangles and vertices. These trees are encoded separately in a lossless manner. The geometry encoding of 3DMC performs lossy compression on the geometric data associated with each vertex location, and possibly, vertex normals, colors, and texture coordinates. The normal encoding technique proposed in MPEG4 3DMC is discussed in this document as a prior-art.

BRIEF DESCRIPTION OF THE DRAWINGS

[004] The accompanying drawings, which are included as part of the present specification, illustrate the presently preferred embodiment of the present invention and together with the general description given above and the detailed description of the preferred embodiment given below serve to explain and teach the principles of the present invention.

Figure 1 illustrates a functional block diagram of a prior art normal encoding technique that adopts a vector quantization (VQ) approach;

Figure 2 illustrates a functional block diagram 200 of prior art vector quantization normal predictive decoding;

Figure 3 illustrates a computer system 2000 representing an integrated multi-processor, in which elements of the present parameterized normal predictive encoder and decoder may be implemented;

Figure 4 illustrates a functional block diagram of parameterized normal predictive encoding according to one embodiment of the present invention;

Figure 5 illustrates a functional block diagram of parameterized normal predictive decoding according to one embodiment of the present invention;

Figure 6 is a flow diagram of the process for parameterized normal predictive encoding; and

Figure 7 is a flow diagram of the process for parameterized normal predictive decoding.

DETAILED DESCRIPTION

[005] A method and system for parameterized normal predictive encoding is disclosed. In one embodiment, the method comprises performing parameterization on three-dimensional graphics model geometric data. Scalar quantization is performed on the three-dimensional graphics model geometric data. The three-dimensional graphics model geometric data is encoded differentially. Coded and compressed three-dimensional graphics model geometric data is generated.

[006] In the following description, for purposes of explanation, specific nomenclature is set forth to provide a thorough understanding of the present invention. However, it will be apparent to one skilled in the art that these specific details are not required in order to practice the present invention.

[007] Some portions of the detailed descriptions which follow are presented in terms of algorithms and symbolic representations of operations on data bits within a computer memory. These algorithmic descriptions and representations are the means used by those skilled in the data processing arts to most effectively convey the substance of their work to others skilled in the art. An algorithm is here, and generally, conceived to be a self-consistent sequence of steps leading to a desired result. The steps are those requiring physical manipulations of physical quantities. Usually, though not necessarily, these quantities take the form of electrical or magnetic signals capable of being stored, transferred, combined, compared, and otherwise manipulated. It has proven convenient at times, principally for reasons of common usage, to refer to these signals as bits, values, elements, symbols, characters, terms, numbers, or the like.

[008] It should be borne in mind, however, that all of these and similar terms are to be associated with the appropriate physical quantities and are merely convenient labels applied to these quantities. Unless specifically stated otherwise as apparent from the following discussion, it is appreciated that throughout the description, discussions utilizing terms such as "processing" or "computing" or "calculating" or "determining" or "displaying" or the like, refer to the action and processes of a computer system, or similar electronic computing device, that manipulates and transforms data represented as physical (electronic) quantities within the computer system's registers and memories into other data similarly represented as physical quantities within the computer system memories or registers or other such information storage, transmission or display devices.

[009] The present invention also relates to apparatus for performing the operations herein. This apparatus may be specially constructed for the required purposes, or it may comprise a general purpose computer selectively activated or reconfigured by a computer program stored in the computer. Such a computer program may be stored in a computer readable storage medium, such as, but is not limited to, any type of disk including floppy disks, optical disks, CD-ROMs, and magnetic-optical disks, read-only memories (ROMs), random access memories (RAMs), EPROMs, EEPROMs, magnetic or optical cards, or any type of media suitable for storing electronic instructions, and each coupled to a computer system bus.

[0010] The algorithms and displays presented herein are not inherently related to any particular computer or other apparatus. Various general purpose systems may be used with programs in accordance with the teachings herein, or it may prove convenient to construct more specialized apparatus to perform the required method steps. The required structure for a variety of these systems will appear from the description below. In addition, the present invention is not described with reference to any particular programming language. It will be appreciated that a variety of programming languages may be used to implement the teachings of the invention as described herein.

MPEG4 3DMC normal encoding

[0011] Figure 1 illustrates a functional block diagram 100 of a prior art normal encoding technique (proposed in MPEG4 3DMC) that adopts a vector quantization (VQ) approach. Each normal is mapped to a codebook index, which is differentially encoded, and then followed by entropy coding. Normalized normals $N_k=(n^{(k)}_x,n^{(k)}_y,n^{(k)}_z)$ 110 can be thought of as a point in one of the 8 octants on a unit radius sphere. Surface normals in general can point in any direction. One assumption of this technique is that surface normals are uniformly distributed over the unit sphere. Since each octant on the sphere is symmetrical, representing the locations of normals in one octant is sufficient to represent any normals in the unit sphere if the octant to which the normals belong is known.

[0012] Since each octant is symmetrical, the choice of the octant where encoding takes place does not matter. Under VQ normal encoding scheme, all the encoding takes place in the positive octant, in the region bounded by:

$$\begin{aligned} x &\geq 0 \\ y &\geq 0 \\ z &\geq 0 \\ x^2 + y^2 + z^2 &= 1 \end{aligned}$$

[0013] At processing block 120, VQ encoding is performed. The absolute values of each coordinate of the points on the sphere are taken, so that each "absolute" normal $N_k' = (n^{(k)'}_x, n^{(k)'}_y, n^{(k)'}_z)$ now lies on the positive octant. Then N_k' is mapped to an index $i_k \in J \equiv \{0,1,2,\ldots 2^{2Nmax}-1\}$, via the mapping (hereafter known as VQ encoding) defined as follows:

$$i_k = 2Mi_y^{(k)} - i_y^{(k)^2} + 2i_x^{(k)} + U_k$$

$$\text{where } M = 2^{N_{\max}}$$

$$i_x^{(k)}{}' = floor(\frac{n_x^{(k)}{}'M}{n_x^{(k)}{}' + n_y^{(k)}{}' + n_z^{(k)}{}'})$$

$$i_y^{(k)}{}' = floor(\frac{n_y^{(k)}{}'M}{n_x^{(k)}{}' + n_y^{(k)}{}' + n_z^{(k)}{}'})$$

$$i_z^{(k)} = floor(\frac{n_z^{(k)}{}'M}{n_x^{(k)}{}' + n_y^{(k)}{}' + n_z^{(k)}{}'})$$

$$i_x^{(k)} = \begin{cases} i_x^{(k)}{}' & i_x^{(k)}{}' \neq M \\ i_x^{(k)}{}' - 1 & i_x^{(k)}{}' = M \end{cases}$$

$$i_y^{(k)} = \begin{cases} i_y^{(k)}{}' & i_x^{(k)}{}' + i_y^{(k)}{}' \neq M \\ i_y^{(k)}{}' - 1 & i_x^{(k)}{}' + i_y^{(k)}{}' = M \end{cases}$$

$$U_k = \begin{cases} 1 & M - i_x^{(k)} - i_y^{(k)} - i_z^{(k)} = 2 \\ 0 & M - i_x^{(k)} - i_y^{(k)} - i_z^{(k)} \neq 2 \end{cases}$$

[0019] The index i_k belonging to the normal N_k', which can be represented by $2N_{max}$ bits, is concatenated with its associated 3-bit octant code, o_k, to form the normal code C_k 130 which is represented by $2N_{max}+3$ bits. Under VQ normal encoding technique, three of the normal representation bits, o_k, are used to encode the octant to which the N_k belongs (determined by the three sign bits of the x,y,z components of N_k). The rest of the normal representation bits, i_k, are used to represent the location of a normal in the octant to which it belongs.

[0020] At processing block 140, C_k 130 is passed into a predictor which generates a predicted normal code C'_k 150 from actual normal codes of previously traversed vertices. At summation block 160, the difference between the actual normal code C_k 130 and the predicted normal code C'_k 150 is generated as E_K 170. At processing block 180, E_k 170is entropy coded and compressed data 190 is output. Like vertex positions, vertex normals in the same neighborhood show geometric similarity. Except at locations such as hard edges, neighboring normals tend to point in similar directions. Such locality characteristics can be exploited again in the encoding of the normal codes obtained above, just as vertex positions. Because proximity in the vertex spanning tree often implies geometric proximity of the corresponding vertices, we only need to encode the difference E_k 170 between the actual normal code C_k 130 of a particular vertex k and the predicted normal codes C'_k 150 constructed from actual normal codes of previously traversed vertices of the vertex spanning tree. These corrective terms E_k 170 have on average smaller magnitude and thus less entropy than absolute codes. The error terms are then entropy coded.

MPEG4 3DMC normal decoding

[0021] Figure 2 illustrates a functional block diagram 200 of a prior art vector quantization normal predictive decoding (proposed in MPEG4 3DMC). Decoding the compressed normal is in principle the reverse of the encoding process. At processing block 210, compressed data 190 is entropy decoded, resulting in the generation of E_k220, representing the error between the actual normal code C_k260 and the predicted normal code C_k240. At summation block 230, E_k, 220 is added to C'_k240 to generate the normal code C_k 260. At processing block 250, a predicted value C'_k240 is generated from the normal code C_k260. At processing block 250, a predicted value C_k 240 is generated from the normal code C_k 260. At processing block 270, VQ decoding is performed on C_k 260 to generate N_k 280. N_k 280 is N_k 110 after being coded and then decoded. The actual normal code C_k 260 corresponding to a particular vertex k is constructed by adding its associated normal code error E_k 220 to the predicted normal codes C'_k 240 obtained from normal codes of previously traversed vertices in the vertex spanning tree.

[0022] The normal code C_k 260 consists of the octant code, o_k, which represents the octant to which the normal N_k 110 belongs, and the index, i_k, to which the normal N_k 110 is mapped. The decoder maps the index i_k back to the reproduction vector $N'_k=(n'^{(k)}_x,n'^{(k)}_y,n'^{(k)}_z)$, called codewords, via the following mapping (hereafter known as VQ decoding):

$$
\begin{aligned}
n'^{(k)}_x &= (3x_k + 1 + U_k)/M \\
n'^{(k)}_y &= (3y_k + 1 + U_k)/M \\
n'^{(k)}_z &= 3 - n'^{(k)}_x - n'^{(k)}_y \\
\text{where } M &= 2^{N_{\max}} \\
y_k &= floor(M - \sqrt{M^2 - i_k}) \\
x_k &= floor((i_k + y_k(y_k - 2M))/2) \\
U_k &= (i_k + y_k(y_k - 2M)) \bmod 2
\end{aligned}
$$

[0023] It can be shown that the above mapping makes sure that for every $i_k \neq i_j, N'_k \neq N'_j$. The octant code o_k is then extracted to restore the sign bits of the reproduction vector N'_k.

[0024] **FIGURE 3** illustrates a computer system 2000 representing an integrated multi-processor, in which elements of the present parameterized normal predictive encoder and decoder may be implemented. One embodiment of computer system 2000 comprises a system bus 2200 for communicating information, and a processor 2100 coupled to bus 2200 for processing information. Computer system 2000 further comprises a random access memory (RAM) or other dynamic storage device 2250 (referred to herein as main memory), coupled to bus 2200 for storing information and instructions to be executed by processor 2100. Main memory 2250 also may be used for storing temporary variables or other intermediate information during execution of instructions by processor 2100. Computer system 2000 also may include a read only memory (ROM) and/or other static storage device 2260 coupled to bus 2200 for storing static information and instructions used by processor 210.

[0025] A data storage device 2270 such as a magnetic disk or optical disc and its corresponding drive may also be coupled to computer system 2000 for storing information and instructions. Computer system 2000 can also be coupled to a second I/O bus 2500 via an I/O interface 2300. A plurality of I/O devices may be coupled to I/O bus 2500, including a display device 2430, an input device (e.g., an alphanumeric input device 2420 and/or a cursor control device 2410). For example, video and graphics clips and related information may be presented to the user on the display device 2430.

[0026] The communication device 2400 is for accessing other computers (servers or clients) via a network. The communication device 2400 may comprise a modem, a network interface card, or other well known interface device, such as those used for coupling to Ethernet, token ring, or other types of networks.

Parameterized normal predictive encoding

[0027] Figure 4 illustrates a functional block diagram 300 of parameterized normal predictive encoding according to one embodiment of the present invention. The functions described herein, may be performed by an apparatus such as that of **Figure 3**. At processing block 320, normal invention is encoded in its parameterized form. Normalized normals $N_k=(n^{(k)}_x,n^{(k)}_y,n^{(k)}_z)$, 310 belonging to a particular vertex k, have unit length and can be mapped easily into spherical coordinates as follows:

$$\phi^{(k)} = \sin^{-1} n_y^{(k)}$$
$$\theta^{(k)} = \tan^{-1}(n_z^{(k)} / n_x^{(k)})$$

[0028] $\theta^{(k)}$ is the angle about the y-axis; $\phi^{(k)}$ is the longitudinal angle from the y=0 plane. Note that $\phi^{(k)}$ goes from $-\pi/2$ to $\pi/2$ and $\theta^{(k)}$ goes from $-\pi$ to π. $\phi^{(k)}$ and $\theta^{(k)}$, are first normalized into the range from –0.5 to 0.5 to form $\phi^{(k)}_n$ and $\theta^{(k)}_n$ and then uniformly quantized to integers $\phi_I^{(k)}$ 321 and $\theta_I^{(k)}$ 322 separately as follows:

$$\theta_I^{(k)} = \left\lfloor \theta^{(k)} \cdot 2^{b_\theta} / 2\pi \right\rfloor$$
$$\phi_I^{(k)} = \left\lfloor \phi^{(k)} \cdot 2^{b_\phi} / \pi \right\rfloor$$

[0029] The mapping from N_k 310 to $\phi_l^{(k)}$ 321 and $\theta_l^{(k)}$ 322 is known hereafter as parameterized normal encoding (PNE). b_ϕ and b_θ are total number of bits used to represent $\phi^{(k)}_n$ and $\theta^{(k)}_n$, or the quantization precisions of $\phi^{(k)}_n$ and $\theta^{(k)}_n$ in number of bits. So $2^{-b\phi}$ and $2^{-b\theta}$ are the quantization steps. Note that since $\phi^{(k)}_n$ and $\theta^{(k)}_n$ can take any value in [-0.5, 0.5], $\phi_l^{(k)}$ 321 can be any b_ϕ-bit integer from $-2^{b\phi-1}$ to $2^{b\phi-1}-1$in 2's complement arithmetic; $\theta_l^{(k)}$ 322 can be any b_θ-bit integer from $-2^{b\theta-1}$ to $2^{b\theta-1}-1$in 2's complement arithmetic.

[0030] $\phi_l^{'(k)}$ 321 is passed onto processing block 330, where the predicted value of $\phi_l^{(k)}$ 321 is generated as $\phi'^{(k)}_l$ 331. $\theta_l^{(k)}$ 322 is passed onto processing block 340, where the predicted value of $\theta_l^{(k)}$ 322 is generated as $\theta'^{(k)}_l$ 341.

At summation blocks 350 and 360, $\phi_l^{(k)}$ 321 and $\theta_l^{(k)}$ 322 are then differentially encoded respectively. To be more specific, the error $E_\phi^{(k)}$ 351 ($E^{(k)}_\theta$) 361 between the actual $\phi_l^{(k)}$ 321 ($\theta_l^{(k)}$ 332) of a particular vertex k and the predicted $\phi_l^{'(k)}$ 331 ($\theta_l^{'(k)}$ 341) constructed from actual ones of previously traversed vertices of the vertex spanning tree. These corrective terms $E^{(k)}_\phi$ 351 and $E^{(k)}_\theta$ 361, again, having on average smaller magnitude and thus less entropy than $\phi_l^{(k)}$ 321 and $\theta_l^{(k)}$ 322, are combined together to be entropy coded at processing block 370. The entropy coding may be Huffman coding, arithmetic coding or Lempel-Ziv coding, although other methods of entropy coding are also contemplated to be within the scope of the present invention. Compressed data is generated after the entropy coding is complete.

Parameterized normal predictive decoding

[0031] Figure 5 illustrates a functional block diagram 400 of parameterized normal predictive decoding according to one embodiment of the present invention. The functions described herein, may be performed by an apparatus such as that of **Figure 3**. Parameterized normal decoding is in principle the reverse of the encoding process. At processing block 420, compressed data 380 is entropy decoded, resulting in the generation of $E^{(k)}{}_{\phi}$ 421 and $E^{(k)}{}_{\theta}$ 422. $E^{(k)}{}_{\phi}$ 421 ($E^{(k)}{}_{\theta}$ 422) represent the error between the predicted values $\phi'^{(k)}_{I}$ 451 ($\theta'^{(k)}_{I}$ 461) and the actual values $\phi_I^{(k)}$ 431 ($\theta_I^{(k)}$ 441) of previously traversed vertices in the vertex spanning tree. At summation block 430, $E_{\phi}^{(k)}$ 421 is added to $\phi^{(k)}$ 451 to generate the parameterized normal code $\phi_I^{(k)}$ 431. At summation block 440, $E_{\theta}{}^{(k)}$ 421 is added to $\theta_I^{(k)}$ 461 to generate parameterized normal code $\theta_I^{(k)}$ 441. At processing blocks 450 and 460, predicted values $\phi'^{(k)}_{I}$ 451 and $\theta'^{(k)}_{I}$ 461 are generated from normal codes $\phi_I^{(k)}$ 431 and $\theta_I^{(k)}$ 441, respectively.

[0032] At processing block 470, $\phi_I^{(k)}$ 431 and $\theta_I^{(k)}$ 441 are then dequantized and unnormalized to give $\phi^{(k)}$ and $\theta^{(k)}$ as follows:

$$\theta^{(k)\prime} = \frac{2\pi\theta_I^{(k)}}{2^{b_\theta}}$$

$$\phi^{(k)\prime} = \frac{\pi\phi_I^{(k)}}{2^{b_\phi}}$$

[0033] The parameterized normal values $\phi^{(k)'}$ and $\theta^{(k)'}$ are then mapped back to rectangular normal coordinates $N'_k=(n'^{(k)}_x,n'^{(k)}_y,n'^{(k)}_z)$ 471as follows:

$$n'^{(k)}_y = \sin\phi^{(k)'}$$
$$n'^{(k)}_x = \cos\theta^{(k)'}\cos\phi^{(k)'}$$
$$n'^{(k)}_z = \sin\theta^{(k)'}\cos\phi^{(k)'}$$

[0034] The mapping from $\phi^{(k)}_l$ and $\theta^{(k)}_l$to N'_k 471 is known hereafter as parameterized normal decoding (PND).

[0035] Figure 6 is a flow diagram of the process for parameterized normal predictive encoding. The process 500 begins at start block 501. At processing block 510, normalized normal N_k 310 is first parameterized and then quantized to give $\phi^{(k)}_l$ 321 and $\theta^{(k)}_l$ 332 using the parameterized normal encoding technique described above. Flow continues to processing block 520, where $\phi_l^{(k)}$321 and $\theta_l^{(k)}$ 322 are differentially encoded, such that the errors $E^{(k)}_\phi$ 351 and $E^{(k)}_\phi$ 361 are formed by subtracting the predicted values $\phi_l'^{(k)}$331 and $\theta_l'^{(k)}$ 341 from the actual ones $\phi_l^{(k)}$321 and $\theta_l^{(k)}$ 322 of previously traversed vertices. At processing block 530, the errors $E^{(k)}_\phi$ 351 and $E^{(k)}_\theta$ 361 are combined together to be entropy coded. Compressed and coded data 380 results. The process completes at block 540.

[0036] Figure 7 is a flow diagram of the process for parameterized normal predictive decoding. The process 600 begins at start block 601. At processing block 610, compressed data 380 is entropy decoded to give $E^{(k)}_\phi$ 421 and $E^{(k)}_\theta$

422. Flow continues to processing block 620, where $E^{(k)}_{\phi}$ 421 and $E^{(k)}_{\theta}$ 422 are differentially decoded. The error values $E^{(k)}_{\phi}$ 421 and $E^{(k)}_{\theta}$ 422 are added to the predicted values $\phi_{l}^{(k)}$ 451 and $\theta_{l}^{(k)}$ 461 of previously traversed vertices in the vertex spanning tree to give the actual values $\phi_{l}^{(k)}$ 431 and $\theta_{l}^{(k)}$ 441. At processing block 630, $\phi_{l}^{(k)}$ 431 and $\theta_{l}^{(k)}$441 are then dequantized, unnormalized and mapped back to the reconstruction normal N_K 447 using the parameterized normal decoding technique described above. The process completes at block 640.

[0037] A method and system for parameterized predictive normal encoding and decoding are disclosed. Generally, a parameterized normal prediction coding scheme in which scalar quantization is done to each individual spherical coordinate of the normals is described. The quantized spherical coordinates are then differentially encoded and entropy coded. Although the present invention has been described with respect to specific examples and subsystems, it will be apparent to those of ordinary skill in the art that the invention is not limited to these specific examples or subsystems but extends to other embodiments as well. The present invention includes all of these other embodiments as specified in the claims that follow.